

# Lateral Movement & Pivoting

Course Introduction



# Alexis Ahmed

Senior Penetration Tester @HackerSploit  
Offensive Security Instructor @INE

---

# Course Topic Overview

- + Introduction To Lateral Movement & Pivoting
- + Windows Lateral Movement Techniques
- + Linux Lateral Movement Techniques
- + Windows Pivoting Techniques
- + Linux Pivoting Techniques

# Prerequisites

- + Basic Understanding of Computer Networking
  - + Knowledge of IP addresses, subnetting, routing, and network devices (switches, routers, firewalls).
  - + Familiarity with common network protocols (TCP, UDP, HTTP, DNS, etc.).
- + Fundamentals of Operating Systems
  - + Basic knowledge of Windows and Linux operating systems, including their command-line interfaces.
  - + Understanding of system processes, file systems, and user permissions.
- + Introductory Knowledge of Cybersecurity Concepts
  - + Awareness of common cybersecurity threats (e.g., phishing, malware, brute-force attacks).
  - + Familiarity with the principles of information security (confidentiality, integrity, availability).
- + Experience with Penetration Testing Tools
  - + Some experience using common penetration testing tools (e.g., Metasploit, Nmap, Wireshark).
  - + Understanding of basic penetration testing methodologies and techniques.

# Learning Objectives:

1. Understanding Lateral Movement and Pivoting
  - Define and differentiate between lateral movement and pivoting in the context of penetration testing.
  - Explain the importance and impact of these techniques in penetration testing and red team operations.
2. Lateral Movement
  - List and describe common lateral movement techniques that can be performed in Windows and Linux environments.
  - Demonstrate competency in the use of various tools and techniques for lateral movement in Windows and Linux environments.
  - Leverage techniques and tools like PsExec, WMI, SSH, RDP, and others to move laterally within Windows and Linux environment.
3. Pivoting
  - Describe how pivoting works and why it's used in red team engagements.
  - Demonstrate competency in using various tools and techniques for network pivoting, including SSH tunneling, VPNs, and SOCKS proxies.
  - Leverage pivoting techniques to access different network segments.
  - Implement multi-hop pivots to navigate complex network topologies Detect and Mitigate Lateral Movement and Pivoting.

**Let's Get Started!**



# Introduction To Lateral Movement & Pivoting

# Lateral Movement

- Lateral movement refers to the process of moving from one compromised system to other systems within a network.
- Penetration testers and red teamers utilize this technique to explore and access additional systems, typically aiming to reach high-value targets or sensitive data.
- The primary goal of lateral movement is to escalate access and privileges, effectively broadening the attacker's control within the network. To do this, attackers exploit vulnerabilities, use compromised credentials, or exploit misconfigurations to gain entry into additional systems.



# Lateral Movement

- Methods of lateral movement include exploiting vulnerabilities in network services, credential dumping to impersonate users, abusing shared resources, or leveraging various forms of remote access (like RDP, SSH, or WMI).
- Why? Let's assume you are on an internal Windows network and you've recovered some valid local/domain credentials. Now what do you actually do with these credentials?
- This is where lateral movement comes in to play, you use these credentials to authenticate to target systems through various remote authentication protocols like SMB, RDP etc

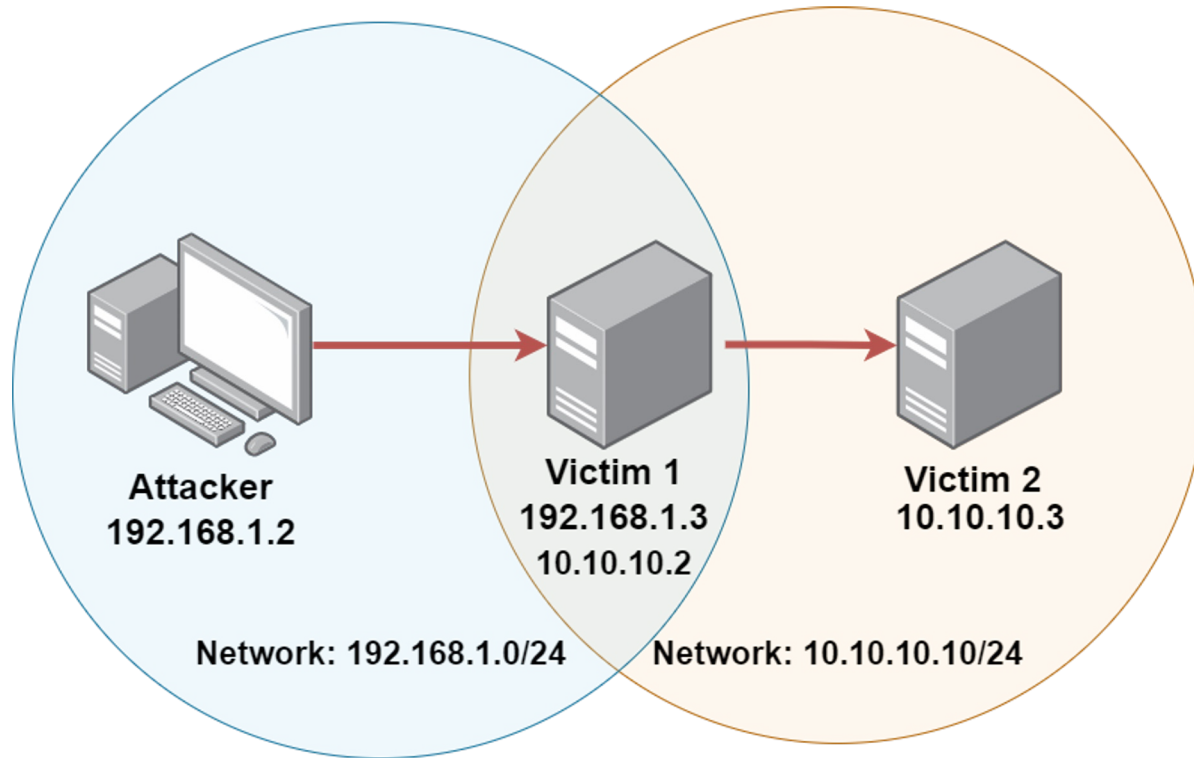
# Pivoting

- Pivoting, while often used interchangeably with lateral movement, has a more specific meaning.
- It involves using a compromised system as a "stepping stone" to access other systems or network segments that are otherwise inaccessible from the attacker's initial entry point.
- Pivoting is common when attacking segmented networks or when an initial entry point is in a less-secure zone, but the target is in a more secure area.

# Pivoting

- By pivoting, attackers can leverage the compromised system's network connections or resources to reach additional systems or networks.
- Techniques for pivoting include setting up port forwarding, using SSH tunnels, exploiting VPNs or other network bridging methods, or configuring proxy servers on the compromised host.

# Pivoting Visualized



# Differences Between Lateral Movement & Pivoting

	Lateral Movement	Pivoting
SCOPE	Lateral movement generally refers to moving within the same network or network segment, targeting adjacent or nearby systems.	Pivoting, on the other hand, typically involves using a compromised system to access other network segments or more restricted areas.
APPROACH	Lateral movement often relies on methods like exploiting common vulnerabilities, using shared resources, or credential-based attacks within a network.	Pivoting usually involves more complex network-routing techniques to gain access to isolated or segmented networks.
OBJECTIVE	lateral movement is often focused on escalating privileges or access within a network	Pivoting is aimed at bypassing network boundaries or accessing different network segments.

# Windows Lateral Movement Techniques

# Windows Lateral Movement Techniques

- Windows lateral movement techniques encompass a range of strategies and methods that attackers use to move laterally within a network, gaining access to additional systems or resources.
- Here is an outline of the various Windows lateral movement techniques, along with common tools or utilities that can be used to execute them.

# Credential-Based Lateral Movement Techniques





# Credential-Based Lateral Movement

- Credential-based lateral movement refers to techniques attackers use to move laterally within a network by obtaining, reusing, or exploiting credentials.
- Credentials can be usernames and passwords, cryptographic hashes, or Kerberos tickets.
- Once attackers have these credentials, they can access other systems, typically without needing additional exploitation or vulnerabilities.
- This form of lateral movement is particularly dangerous because it relies on valid credentials, which can make it difficult to detect and often mimics legitimate user behavior.

# Credential-Based Lateral Movement Techniques

## Pass-the-Hash (PtH)

- Attackers use stolen NTLM hash values to authenticate with other systems without knowing the plaintext password.
- Tools: Mimikatz, Metasploit, Impacket (e.g., psexec.py), CrackMapExec.

## Pass-the-Ticket (PtT)

- Attackers use captured Kerberos tickets to authenticate and access resources across a network.
- Tools: Mimikatz, Rubeus, Kerberos exploitation tools.

## Credential Reuse

- Attackers use captured plaintext passwords or hashes to gain access to additional systems.
- Tools: Mimikatz, Metasploit, CrackMapExec, Impacket.

# Credential-Based Lateral Movement Techniques

## Golden/Silver Tickets

- Attackers forge Kerberos tickets to gain long-term or specific access to domain resources.
- Tools: Mimikatz, Rubeus.

# Windows Remote Management Protocols



# Windows Remote Management Protocols

- Remote management protocols play a significant role in Windows administration, allowing administrators to manage systems remotely, troubleshoot, and execute scripts.
- However, these protocols also present opportunities for attackers to move laterally within a network once they've compromised a system or obtained valid credentials.

# Windows Remote Management Protocols

Protocol	Description	Ports	Attack Methods
Windows Remote Management (WinRM)	WinRM is Microsoft's implementation of the WS-Management protocol, allowing remote management and command execution on Windows systems.	Typically operates on ports 5985 (HTTP) and 5986 (HTTPS).	Attackers can use valid credentials to access WinRM and execute remote PowerShell commands or scripts. Tools like PowerShell remoting, Invoke-Command, and Impacket's smbexec.py can exploit WinRM for lateral movement.
Remote Desktop Protocol (RDP)	RDP provides a graphical interface for remote access to Windows systems. It is widely used by administrators for remote control and management.	Default port is 3389/TCP	Attackers with valid RDP credentials or who can bypass security controls (like Network Level Authentication) can use RDP for lateral movement. Attackers might also use RDP brute-force attacks to gain access.
Windows Management Instrumentation (WMI)	WMI is a framework for managing Windows systems, allowing remote administration, data collection, and remote command execution.		Attackers can use WMI to execute commands on remote systems, gathering information or performing lateral movement. Tools like Impacket's wmiexec.py, PowerShell, and Metasploit's WMI modules are often used to exploit WMI for remote execution.

# Windows Remote Management Protocols

Protocol	Description	Ports	Attack Methods
Server Message Block (SMB)	SMB is a network protocol for file sharing, printer sharing, and inter-process communication. It is also used for various remote management functions.	Default ports are 445 and 139.	Attackers might use SMB to deploy tools like PsExec for remote code execution, or use tools like CrackMapExec and Impacket's psexec.py to gain remote access and move laterally.

# **Remote Management Protocols - Authenticated Remote Code Execution**





# Authenticated Remote Code Execution

- Remote execution techniques for lateral movement involve running commands or code on a remote system, allowing attackers to move laterally across a network.
- These techniques are critical to understand in penetration testing, incident response, and cybersecurity as they enable attackers to access additional systems and potentially escalate privileges.
- Remote execution techniques can be based on a variety of methods, such as using network protocols, remote management tools, or leveraging shared resources.

# Authenticated RCE Techniques

## Remote Desktop Protocol (RDP)

- RDP allows a user to remotely connect to a Windows system with a graphical user interface. Attackers can use compromised credentials or brute force to gain access.
- Attackers with valid RDP credentials can connect to remote systems and perform actions as if they were physically present.
- Tools: xfreerdp, Metasploit modules for RDP, RDP brute force tools.

## Windows Management Instrumentation (WMI)

- WMI is a framework for managing and monitoring Windows systems. It allows remote execution of commands and scripts.
- Attackers use WMI to run code or scripts on remote systems, often for reconnaissance or remote command execution.
- Tools: PowerShell, Impacket (wmiexec.py), Metasploit modules.

# Authenticated RCE Protocols & Techniques

## PowerShell Remoting

- PowerShell provides capabilities for remote administration and scripting. PowerShell remoting allows execution of commands on remote systems.
- Attackers use PowerShell to execute scripts on other systems, potentially to install backdoors, exfiltrate data, or perform other malicious actions.
- Tools: PowerShell (Invoke-Command, Enter-PSSession), Metasploit modules, Empire framework.

## PsExec and Similar Tools

- PsExec and similar tools (like psexec.py from Impacket) allow remote execution over SMB by creating a service on the target system to run commands.
- Attackers can execute code remotely using these tools, often with elevated privileges, to move laterally across the network.
- Tools: PsExec, Impacket (psexec.py), CrackMapExec.

# Lateral Movement With PsExec

# Understanding PsExec



# PsExec

- Authenticated Windows lateral movement via SMB with PsExec refers to the process of using valid credentials to move laterally across a Windows network by remotely executing code on other systems via Server Message Block (SMB).
- PsExec is a utility from the Sysinternals Suite that allows administrators to execute commands on remote systems. However, attackers can also use it for malicious purposes.

# What is PsExec?

- PsExec is a utility from the Sysinternals Suite, designed by Mark Russinovich, that allows users to execute commands on remote Windows systems.
- It is widely used by system administrators for remote management and troubleshooting. However, due to its capability to execute commands remotely with administrative privileges, PsExec has also become a popular tool for attackers to facilitate lateral movement in Windows networks.

# How PsExec Works

## 1 - Connection Over SMB

- PsExec establishes a connection to a remote system using Server Message Block (SMB).
- It typically requires credentials for authentication, either as plaintext passwords or NTLM hashes.

## 2 - Named Pipe

- PsExec creates a named pipe on the remote system to facilitate communication between the local PsExec client and the remote service.



# How PsExec Works

## **3 - Temporary Service**

- To execute commands on the remote system, PsExec creates a temporary Windows service.
- This service runs with elevated privileges, allowing it to execute commands or scripts as a system administrator.

## **4 - Execution and Cleanup**

- Once the command or script is executed, PsExec cleans up by removing the temporary service. However, traces like logs or artifacts may remain, providing a trail of activity.

# PsExec - Permissions & Privileges

To authenticate via SMB and execute commands with PsExec, the user account must have appropriate permissions. Here's what this typically means:

## Administrative Privileges

- PsExec typically requires administrative privileges to function properly. This means that the user account used to authenticate over SMB must have the rights to:
  - Create and start a service on the remote system.
  - Access the IPC\$ share, which is used to establish the SMB connection.
  - Read and write to certain directories or system areas.
  - Local Users and Domain Users: PsExec can use both local and domain-based accounts to authenticate over SMB. In domain environments, users with administrative rights in Active Directory may have broader access to remote systems.

# SMB Authentication



# SMB

- SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals (printers and serial ports) between computers on a local network (LAN).
- SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.
- SAMBA is the open source Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.

# SMB Authentication

- The SMB protocol utilizes two levels of authentication, namely:
  - + User Authentication
  - + Share Authentication
- User authentication - Users must provide a username and password in order to authenticate with the SMB server in order to access a share.
- Share authentication - Users must provide a password in order to access restricted share.

**Note: Both of these authentication levels utilize a challenge response authentication system.**



# NTLM Authentication

- NTLM (NT LAN Manager) is a legacy authentication protocol used in Microsoft Windows environments. Although it has been largely replaced by Kerberos in domain-based environments, NTLM is still used in various contexts, especially in environments with older systems or in specific use cases where Kerberos is not feasible.
- Understanding how NTLM authentication works is critical, particularly in the context of Server Message Block (SMB), because it is often a target for attacks, such as Pass-the-Hash.

# NTLM Authentication Process

NTLM authentication operates using a challenge-response mechanism. Here's a breakdown of the NTLM authentication process in the context of SMB:

## **1 - Connection Request**

- A client, such as a Windows system or an application, initiates a connection to an SMB server. This could be for accessing a shared folder, printer, or other SMB-based resources.

## **2 - Server Challenge**

- The SMB server responds with an NTLM "challenge." This challenge is a random value used to ensure that the authentication process involves a unique component for each session.

# NTLM Authentication Process

## 3 - Client Response

- The client calculates a response to the server's challenge. This calculation involves encrypting the challenge using the NTLM hash derived from the user's password. This process ensures that the client has access to the correct NTLM hash without transmitting the plaintext password.
- There are two key parts to the response:
  - NTLMv1 Response: Uses a DES-based mechanism to generate the response. It has been deprecated due to security weaknesses.
  - NTLMv2 Response: More secure, involving a combination of the server's challenge and a client challenge (a unique value generated by the client), providing additional security against replay attacks.

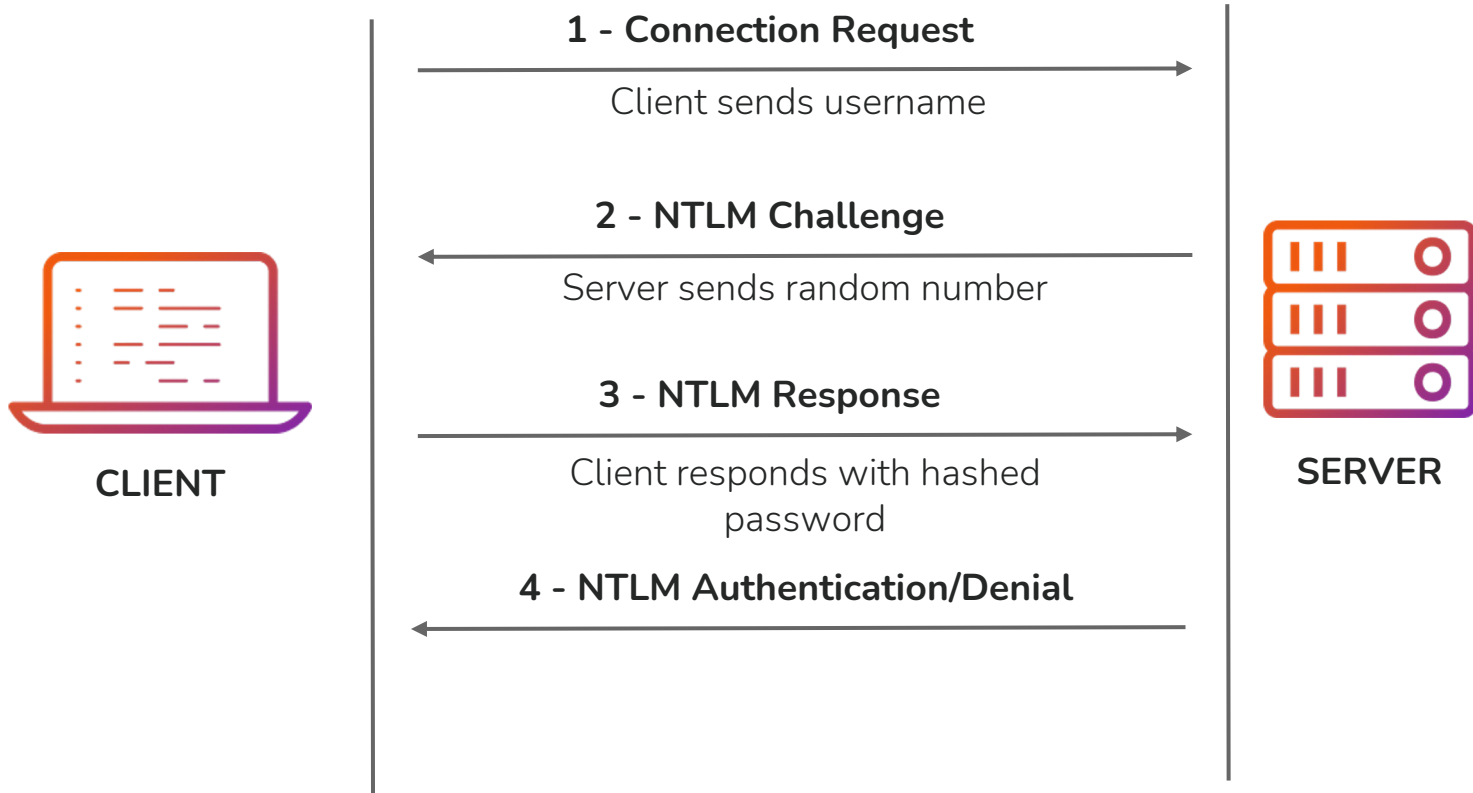


# NTLM Authentication Process

## 4 - Server Verification

- The server verifies the client's response by comparing it to the expected response, derived from its stored NTLM hashes.
- If they match, the client is authenticated, allowing access to SMB resources.

# NTLM Authentication



# Lateral Movement With PsExec



# SMB Authentication via PsExec

- In order to utilize PsExec to gain access to a Windows target, we will need to identify legitimate user accounts and their respective passwords or password hashes.
- This can be done by leveraging various tools and techniques, however, we also have the option to perform username enumeration and consequently password spraying.
- We can narrow down our password spraying attack to only target common, ever present local Windows user accounts like the Administrator, Admin etc
- After we have obtained a legitimate user account and password, we can use the credentials to authenticate with the target system via PsExec and execute arbitrary system commands or obtain a reverse shell.

# Impacket's Implementation of PsExec

- Impacket is a collection of Python scripts and classes designed to interact with network protocols, particularly those used in Windows environments like SMB, DCOM, and others.
- Impacket provides a Python implementation of PsExec, allowing Python-based tools to perform similar operations.

# Impacket's Implementation of PsExec

## psexec.py in Impacket

- This script in Impacket mimics the functionality of PsExec, allowing remote execution of commands on Windows systems. Here's how it typically works:
  - SMB Connection: psexec.py establishes a connection to the remote system using SMB. It uses valid credentials (plaintext or hashed) to authenticate.
  - Creating a Named Pipe: Once connected, the script creates a named pipe to communicate with the remote system.
  - Installing a Service: It installs a temporary service on the remote system to execute the desired command.
  - Executing Commands: The service executes the provided command or script, returning the output via the named pipe.
  - Cleaning Up: After execution, the script removes the temporary service to minimize traces.

# Lateral Movement With PsExec

## 1 - Obtain Credentials

- Attackers acquire valid credentials through methods like credential dumping, phishing, or other means. These credentials may be plaintext passwords or NTLM hashes.

## 2 - Establish SMB Connection

- Using PsExec, the attacker establishes a connection to the target system over SMB. This typically involves providing a username and password, or NTLM hash.

## 3 - Remote Command Execution

- Once connected, PsExec creates a named pipe on the target system to communicate with the PsExec client.
- It then creates a temporary Windows service to execute the desired commands, scripts, or programs.

# Lab Demo: Lateral Movement With PsExec



## References & Resources

- PsExec Documentation:  
<https://learn.microsoft.com/en-us/sysinternals/downloads/psexec#introduction>
- NTLM User Authentication:  
<https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/ntlm-user-authentication>
- Impacket's Python implementation of PsExec:  
<https://github.com/fortra/impacket/blob/master/examples/psexec.py>

A man with glasses and a beard is shown in profile, looking at a computer monitor. The monitor displays a dark-themed code editor with green and blue highlights. The background is dark, and the overall lighting is dim, focusing on the man and his work.

# Lateral Movement With SMBExec

# SMBExec

- SMBExec is a tool used for remote command execution on Windows systems over the Server Message Block (SMB) protocol.
- It is designed to allow administrators (or attackers) to execute commands on remote Windows machines, typically with administrative privileges, using SMB.
- While similar in functionality to PsExec, SMBExec has certain distinctions that make it a unique tool in its own right.

# Key Characteristics of SMBExec

## Remote Command Execution

- SMBExec allows users to execute commands or scripts on remote Windows systems over SMB.

## Authentication via SMB

- The tool connects to remote systems using SMB, typically requiring valid credentials for authentication.

# Key Characteristics of SMBExec

## Does Not Create a Temporary Service

- Unlike PsExec, which creates a temporary Windows service to execute commands, SMBExec operates differently, often using Windows Management Instrumentation (WMI) or similar methods to achieve remote execution without creating additional services.

## Privilege Escalation and Lateral Movement

- SMBExec can be used for administrative tasks or, in a malicious context, for privilege escalation and lateral movement within a network.

# Lab Demo: Lateral Movement With SMBExec



# Lateral Movement With CrackMapExec

# CrackMapExec

- CrackMapExec (CME) is a powerful open-source penetration testing tool designed for security professionals and ethical hackers.
- It is used to enumerate and assess networks, particularly Windows environments, and has extensive capabilities for network reconnaissance, lateral movement, and remote exploitation.




# CrackMapExec Use Cases

- Network Enumeration and Reconnaissance: CME can scan networks to identify hosts, enumerate shares, and gather other valuable information.
- Credential Testing and Brute Forcing: CME can test credentials against network resources, checking for weak or reused passwords. This can help penetration testers identify potential vulnerabilities.
- Lateral Movement: Using various techniques, CME can move laterally across a network, exploiting valid credentials, hashes, or Kerberos tickets.

# CrackMapExec Use Cases

- Privilege Escalation: CME can attempt to escalate privileges on remote systems, providing deeper access to resources.
- Remote Command Execution: CME can execute commands on remote systems over SMB, WMI, or other protocols, allowing for flexible remote management.



# Lab Demo: Lateral Movement With CrackMapExec

# Lateral Movement Via RDP



# Lab Demo: Lateral Movement Via RDP

# Lateral Movement Via WinRM

# WinRM

- Windows Remote Management (WinRM) is a Windows remote management protocol that can be used to facilitate remote access with Windows systems over HTTP/HTTPS.
- Microsoft implemented WinRM in to Windows in order to make life easier for system administrators.
- WinRM is typically used in the following ways:
  - Remotely access and interact with Windows hosts on a local network.
  - Remotely access and execute commands on Windows systems.
  - Manage and configure Windows systems remotely.
- WinRM typically uses TCP port 5985 and 5986 (HTTPS).

# WinRM Authentication

- WinRM implements access control and security for communication between systems through various forms of authentication.
- WinRM authentication involves verifying the identity of a client (such as a remote user or script) attempting to connect to a WinRM service on a Windows system.
- There are several authentication mechanisms used by WinRM, each with different security characteristics.



# WinRM Authentication

## NTLM Authentication

- Challenge-Response: NTLM (NT LAN Manager) authentication operates using a challenge-response mechanism. The client receives a challenge from the server and responds with an encrypted value derived from the NTLM hash.
- Used in Non-Domain Environments: NTLM is typically used in workgroup environments or when Kerberos is not feasible.
- Less Secure than Kerberos: NTLM is more prone to attacks like Pass-the-Hash, making it less secure than Kerberos.

# WinRM Authentication

## Basic Authentication

- Basic authentication sends credentials (username and password) in Base64-encoded format. This method should only be used over HTTPS to ensure encrypted communication.
- Less Secure: Because credentials are sent in a basic format, this method is less secure compared to Kerberos and NTLM.
- Typically Disabled by Default: Basic authentication is often disabled by default due to the obvious security risks.

# WinRM Privileges

- To access Windows Remote Management (WinRM), a user account requires certain privileges and permissions. The specific privileges depend on what operations the user is expected to perform with WinRM and whether the environment is configured to restrict access to certain groups or users.
- Here's a breakdown of the necessary user account privileges for accessing WinRM:
  - Default Configuration and Access
  - By default, WinRM is configured to allow access to users who belong to the local "Administrators" group on a Windows system.
  - WinRM might also be accessible to users who have been granted specific permissions or rights, depending on configuration and policy settings.

# WinRM Privileges

- The **"Remote Management Users"** group is a built-in group in Windows designed to facilitate controlled access to Windows Remote Management (WinRM).
- Members of this group have permission to connect to WinRM-enabled systems and execute specific management tasks without requiring full administrative privileges.

# WinRM Tools



# Evil-WinRM

- Evil-WinRM is an open-source tool designed for interacting with Windows Remote Management (WinRM) services on remote Windows systems.
- It is primarily used by penetration testers and red teamers for post-exploitation activities, such as executing commands, transferring files, and gathering information on remote systems.
- Evil-WinRM is a popular choice for security professionals seeking to assess the security of Windows environments, especially in contexts where WinRM is enabled and accessible.

# CrackMapExec & WinRM

- CrackMapExec (CME) supports several network protocols, with Windows Remote Management (WinRM) being one of them.
- The WinRM functionality in CrackMapExec is particularly useful for assessing the security of Windows environments, allowing testers to interact with remote systems using various authentication methods.
- CrackMapExec can execute PowerShell commands or scripts on remote systems via WinRM. This can be used for tasks like gathering system information, creating users, or running custom scripts.

# PowerShell Remoting (PSRemoting)

- PowerShell Remoting, often referred to as PSRemoting, is a feature of PowerShell that allows you to run PowerShell commands or scripts on remote computers.
- This capability is built on Windows Remote Management (WinRM), a technology designed to facilitate remote management and command execution in Windows environments.
- In the context of lateral movement, PSRemoting can be used to access and execute commands on remote systems, making it a valuable tool for both administrators and attackers.



# Choosing The Right Tool

## For Administrative Tasks:

- Use PSRemoting: If you need a native PowerShell approach for remote management, automation, or administration, PSRemoting is the best choice.

## For Penetration Testing or Red Teaming:

- Use Evil-WinRM: If your focus is on WinRM-based penetration testing, Evil-WinRM is a specialized tool designed for this purpose.
- Use CrackMapExec: If you need broader functionality, including support for multiple protocols and credential testing, CME offers a versatile solution for penetration testing and network reconnaissance.

# Lab Demo: Lateral Movement Via WinRM

## References & Resources

- Windows Remote Management Reference: <https://learn.microsoft.com/en-us/windows/win32/winrm/portal>
- Evil-WinRM: <https://github.com/Hackplayers/evil-winrm>
- PSRemoting: <https://learn.microsoft.com/en-us/powershell/scripting/learn/ps101/08-powershell-remoting?view=powershell-7.4>

# Pass-the-Hash With Metasploit

# Pass-the-Hash (PtH)

- Pass-the-Hash (PtH) is a technique used in lateral movement attacks where an attacker uses a hashed version of a password (commonly an NTLM hash) to authenticate without needing the plaintext password.
- This technique is especially relevant in Windows environments, where NTLM authentication is still supported for compatibility with older systems or non-domain environments.
- Pass-the-Hash allows attackers to impersonate users, gain unauthorized access, and move laterally across a network.

# Pass-the-Hash (PtH) Attacks

## 1 - Obtaining Hashes

- Attackers acquire NTLM hashes through various means, such as credential dumping, memory scraping, or capturing network traffic.
- Tools like Mimikatz, Hashcat, or Metasploit can be used to extract or manipulate NTLM hashes from compromised systems.

## 2 - Using NTLM Hashes for Authentication

- Once the attacker has an NTLM hash, they can use it to authenticate with other systems or services without needing the plaintext password.
- The hash is used to respond to an NTLM challenge, effectively bypassing the need for a plaintext password.

# Pass-the-Hash (PtH) Attacks

## 3 - Connecting to Remote Systems

- Attackers can use the NTLM hash to establish connections with remote systems. Common methods include using Server Message Block (SMB), Windows Management Instrumentation (WMI), or Remote Desktop Protocol (RDP).
- Tools like CrackMapExec, PsExec, or Impacket allow attackers to authenticate and execute commands on remote systems using Pass-the-Hash.

## 4 - Lateral Movement

- Using Pass-the-Hash, attackers can move laterally across a network, connecting to other systems and performing various tasks, such as remote command execution, data exfiltration, or privilege escalation.
- This technique is particularly effective because it relies on legitimate authentication mechanisms, often blending in with normal network activity.

# Lab Demo: Pass-the-Hash With Metasploit



# Pass-the-Hash With WMIExec

# WMI

- Windows Management Instrumentation (WMI) is a Microsoft framework for managing and monitoring Windows-based systems.
- It allows administrators to perform various tasks, such as gathering system information, executing scripts, and controlling system operations, all through a consistent interface.
- WMI can be accessed locally on a Windows system or remotely over a network, providing flexibility for system administration.

# WMI

- WMI relies on underlying protocols for remote communication, primarily Distributed Component Object Model (DCOM) and Remote Procedure Call (RPC).
- **Port 135:** This is the RPC Endpoint Mapper, used to discover other RPC-based services on a remote system. WMI uses this port to initiate remote communication.
- **Dynamic RPC Ports:** After connecting to port 135, the communication is redirected to a dynamically assigned RPC port. These dynamic ports typically fall within the range of 49152–65535, though this range can be customized through Group Policy or registry settings.

# WMIExec

- **wmiexec** is a technique used to execute commands or scripts on remote Windows systems via Windows Management Instrumentation (WMI).
- In the context of lateral movement, wmiexec provides a way for attackers or penetration testers to interact with remote systems without requiring additional services or software installations.
- This approach allows attackers to move laterally across a network, executing code remotely to gain further access or escalate privileges.

# How WMIExec Works

- **WMI:** Windows Management Instrumentation is a framework that allows interaction with system components and management of various tasks on Windows systems. WMI is commonly used for remote management and administration.
- **DCOM and RPC:** WMI operates over Distributed Component Object Model (DCOM) and Remote Procedure Call (RPC), allowing remote communication with Windows systems.
- **Remote Command Execution:** wmiexec uses WMI to remotely execute commands or scripts on a target system. It does so by sending a WMI request to execute a command, and then retrieves the output from the remote system.

# Lateral Movement With WMIExec

## Remote Command Execution

- wmiexec allows execution of commands on remote systems, providing a method to interact with and manipulate the target system.

## Privilege Escalation

- wmiexec can be used to execute commands with elevated privileges, potentially allowing attackers to escalate their access rights on the remote system.

## Minimal Footprint

- wmiexec does not require additional software installations or creating new services. This minimal footprint makes it less detectable and can bypass certain security controls.
- The ability to execute commands without leaving extensive traces makes it attractive for lateral movement.

# Lateral Movement With WMIExec

- Several tools incorporate wmiexec functionality, allowing attackers or penetration testers to execute commands via WMI. Here are some examples:
  - Impacket: A popular collection of Python tools for interacting with Windows network protocols. Impacket's wmiexec.py allows remote command execution via WMI.
  - CrackMapExec: A multi-functional penetration testing tool that includes WMI-based lateral movement capabilities.
  - Custom Scripts: Attackers and penetration testers may create custom scripts that leverage WMI for lateral movement.

# Lab Demo: Pass-the-Hash With WMIExec






# Linux Lateral Movement Techniques

# Lab Demo: Linux Lateral Movement Techniques

# Pivoting & Port Forwarding With Metasploit



# Lab Demo: Pivoting & Port Forwarding With Metasploit

# Pivoting With SOCKS Proxy



# Lab Demo: Pivoting With SOCKS Proxy

# Pivoting Via SSH Tunneling

# SSH Tunneling

- SSH tunneling, also known as SSH port forwarding, is a technique that uses Secure Shell (SSH) to create encrypted tunnels for network traffic.
- It allows users to securely forward data through an SSH connection, providing a way to protect sensitive information from interception and to bypass certain network restrictions.
- SSH tunneling is a versatile tool used for remote access, secure communication, and overcoming firewalls or network segmentation.



# Pivoting Via SSH Tunneling

- SSH tunneling, in the context of pivoting, refers to a technique used to create secure tunnels through Secure Shell (SSH) connections that allow attackers or penetration testers to access other systems or network segments.
- This method is often employed when an initial foothold has been established, and further access to restricted resources or lateral movement across a network is needed.
- SSH tunneling creates an encrypted tunnel between a local and a remote system, allowing you to forward network traffic securely through this tunnel. In pivoting, SSH tunneling can be used to connect to additional systems, establish secure communication channels, or gain access to restricted network segments.

# SSH Tunneling Techniques

## Local Port Forwarding

- Local port forwarding redirects traffic from a local port on the client system to a specified port on the remote system. This allows you to create secure tunnels to internal resources.
- **Use Case for Pivoting:** Local port forwarding can be used to connect to internal services on a compromised system.
- This can facilitate access to other network segments or systems behind firewalls.

# SSH Tunneling Techniques

## Remote Port Forwarding

- Remote port forwarding allows traffic from a port on the remote system to be forwarded to a specified port on the local system. This technique is useful for enabling remote access or establishing backdoors.
- **Use Case for Pivoting:** Remote port forwarding can be used to create a tunnel that allows remote systems to connect to internal resources via a compromised system.
- This can be used to maintain persistence or access internal networks from outside the firewall.

# SSH Tunneling Techniques

## Dynamic Port Forwarding

- Dynamic port forwarding creates a SOCKS proxy, allowing flexible port forwarding based on client requests. This is useful for tunneling various types of traffic through the SSH connection.
- **Use Case for Pivoting:** Dynamic port forwarding can be used to create a SOCKS proxy, enabling flexible tunneling.
- This can be used to access multiple internal resources or services through a single SSH connection, providing a broader scope for pivoting.



# Lab Demo: Pivoting Via SSH Tunneling

# Pivoting With reGeorg

# Lab Demo: Pivoting With reGeorg

# Lateral Movement & Pivoting

Course Conclusion



# Learning Objectives:

1. Understanding Lateral Movement and Pivoting
  - Define and differentiate between lateral movement and pivoting in the context of penetration testing.
  - Explain the importance and impact of these techniques in penetration testing and red team operations.
2. Lateral Movement
  - List and describe common lateral movement techniques that can be performed in Windows and Linux environments.
  - Demonstrate competency in the use of various tools and techniques for lateral movement in Windows and Linux environments.
  - Leverage techniques and tools like PsExec, WMI, SSH, RDP, and others to move laterally within Windows and Linux environment.
3. Pivoting
  - Describe how pivoting works and why it's used in red team engagements.
  - Demonstrate competency in using various tools and techniques for network pivoting, including SSH tunneling, VPNs, and SOCKS proxies.
  - Leverage pivoting techniques to access different network segments.
  - Implement multi-hop pivots to navigate complex network topologies Detect and Mitigate Lateral Movement and Pivoting.

**Thank You!**



*EXPERTS AT MAKING YOU AN EXPERT*

